Package ‘kmlcov’

February 20, 2015

Type  Package
Title  Clustering longitudinal data using the likelihood as a metric of distance
Version  1.0.1
Date  2013-08-21
Description  ‘kmlcov’ Cluster longitudinal data using the likelihood as a metric of distance. The generalised linear model allow the user to introduce covariates with different level effects (2 levels).
License  GPL-2
Depends  methods
Author  Mamoun O. Benghezal [aut, cre],
        Christophe Genolini [ctb]
Maintainer  Mamoun O. Benghezal <mobenghezal@gmail.com>
NeedsCompilation  no
Repository  CRAN
Date/Publication  2013-08-21 16:07:27

R topics documented:

  kmlcov-package .................................................. 2
  addIndic .......................................................... 3
  affect_rand ....................................................... 4
  artifdata .......................................................... 5
  Converge-class ................................................... 5
  Converge-methods ............................................... 5
  getNomCoef ....................................................... 6
  glmClust .......................................................... 7
  GImCluster-class ............................................... 9
  GImCluster-methods ............................................. 10
  kmlCov ........................................................... 10
kmlcov-package

kmlcov-package

Clustering longitudinal data using the likelihood as a metric of distance

Description

'kmlcov' Cluster longitudinal data using the likelihood as a metric of distance. The generalised linear model allow the user to introduce covariates with different level effects (2 levels).

Details

Package: kmlcov
Type: Package
Version: 1.0.1
Date: 2013-08-21
Authors@R: c(person("Mamoun O", "Benghezal", role = c("aut", "cre"), email = "mobenghezal@gmail.com"), person("Christophe", "Genolini", role = c("ctb"), email = "christophe.genolini@u-paris10.fr"),
License: GPL-2
Depends: methods
Collate: 'functions4glmClust.R' 'GlmCluster.R' 'glmclust-internal.R' 'glmClust.R' 'kmlcov-package.R' 'kmlCov.R'

Index:

glmClust Clustering longitudinal data
kmlCov Clustering longitudinal data from different starting conditions
which_best Seek the best partitions

Converge-class Class "Converge"
GlmCluster-class Class "GlmCluster"
KmlCovList-class Class "KmlCovList"
addIndic Create the new formula with the indicator covariates
affect_rand Affect randomly the individuals to the clusters
getNomCoef Get the name of the coefficients in the 'glm' object according to the current cluster
Overview

To cluster longitudinal data, `kmlcov` implement an ECM type algorithm which assign the trajectories to the cluster which maximise the likelihood. It is possible to introduce covariates via the generalised linear model with different level effects (2 levels) all specified in one formula. The package implements the `plot` function to produce the diagrams at the condition of not having more than 2 different effects (although the program can deal with more than two effects) for e.g. time and treatment or time and sex q.v. the help of linkglmClust or `kmlCov`. To plot the main trajectories with more than two effects we recommand to use `ggplot` of the package ggplot2.

To cluster longitudinal data, 2 functions have to be remembered `glmClust` and `kmlCov`, the first run the algorithm for clustering one time and the second run the same algorithm multiple times with different starting conditions. The method is greatly sensitive to the initial conditions, we therefore recommand to use `kmlCov` although it takes much more time.

See Also

- `kmlCov`
- `glmClust`
- `which_best`

Description

Write the new formula given the covariates with a level cluster effect, the number of clusters and the type parametric on time or not.
Usage

\texttt{addIndic(covar, nClust, parametric = TRUE, nomClust = 'G')}  

Arguments

covar \hspace{0.5cm} A 'character' vector of covariates.  
nClust \hspace{0.5cm} Number of clusters.  
parametric \hspace{0.5cm} If [TRUE] it means we are parametric on time.  
nomClust \hspace{0.5cm} The beginning of the name of the indicator covariates 'Ga, Gb, .., etc.

Details

Given the covariates and the number of clusters, it returns a character string which will be converted inside \texttt{glmClust} to a formula to represent the covariates with different cluster effect.

Value

A character string which will be used as a 'formula'.

Note

Meant to be used internally.

\begin{verbatim}
af\textunderscore fect\_rand
\end{verbatim}

\textbf{Description}

\textit{Affect randomly the individuals to the clusters.}

Usage

\texttt{affect\_rand(nObs, nClust)}

Arguments
	nObs \hspace{0.5cm} Number of observations.  
nClust \hspace{0.5cm} Number of clusters.

Details

\textit{Affect randomly the individuals to the clusters providing no empty clusters.}

Value

A [vector] of length 'nObs' containing the affectation to the clusters.
artifdata

**Note**

Meant to be used internally.

---

| artifdata | Artificial data |

**Description**

'artifdata' contains artificial data obtained with the `rnorm` function and should contain 4 cluster of trajectories.

**Format**

A `[data.frame]` containing 500 measures of 50 individuals (trajectories) identified by a column 'id', and the associated taking 'time', 'time2' and 'time3' of some drug for example. In additional there are 2 more columns, 'treatment' is a binary column indicating for example individuals receiving a high dose of some drug and the other receiving a normal dose (coded by 0), 'treatTime' is the 'time' column multiplied by 'treatment'.

---

**Converge-class**

**Class** "Converge"

**Description**

Contain informations about the convergence and the number of iterations of the algorithm.

**Slots**

- `nIter`: Number of iterations of the algorithm.
- `convergence`: "logical" if [TRUE] then the algorithm met convergence.

**Note**

Meant to be used internally.

---

**Converge-methods**

**Method for function** `show`

**Description**

Print a message about the convergence and the number of iterations of the algorithm.
getNomCoef  

Get the name of the coefficients in the 'glm' object according to the current cluster

Description

This function creates and return a vector containing the name of the coefficients associated to the current cluster.

Usage

getNomCoef(covar, cov_fix, nomClust, itrClust, parametric = TRUE)

Arguments

covar       A vector of [character] indicating the covariates with a levec cluster effect.
cov_fix     A vector of [character] indicating the covariates with the same effect in each cluster.
nomClust    The beginning of the name of the undicator covariates, by default 'G[letters]'.
itrClust    The number of the current cluster.
parametric  By default [TRUE] for parametric on time.

Details

Given the name of the covariates and the number of the current cluster, it constructs a vector used to retrieve the coefficients from a 'glm' object, these coefficients are used to calculate the predicted values of the current cluster.

Value

A vector of [character] giving the name of the coefficients associated to a with a given cluster.

Note

Meant to be used internally.
glmClust

Description

'glmClust' cluster longitudinal data (trajectories) using the likelihood as a metric of distance, it also
deals with multiples covariates with different effects using the generalised linear model 'glm'.

Usage

glmClust(formula, data, ident, timeVar, nClust, family =
'gaussian', effectVar = '', weights =
rep(1,nrow(data)), affUser, timeParametric = TRUE,
separateSampling = TRUE, max_itr = 100, verbose = TRUE)

Arguments

formula A symbolic description of the model. In the parametric case we write for ex-
ample 'y ~ clust(time+time2) + pop(sex)', here 'time' and 'time2' will have a
different effect according to the cluster, the 'sex' effect is the same for all the
clusters. In the non-parametric case only one covariate is allowed.
data A [data.frame] in long format (no missing values) which means that each line
corresponds to one measure of the observed phenomenon, and one individual
may have multiple measures (lines) identified by an identity column. In the
non-parametric case the totality of patients must have all the measurements at
fixed times.
nClust The number of clusters, between 2 and 26.
ident Name of the column identity in the data.
timeVar Name of the 'time' column in the data.
family A description of the error distribution and link function to be used in the model,
by default 'gaussian'. This can be a character string naming a family function, a
family function or the result of a call to a family function. (See family for more
details of family functions).
effectVar Name of the effect specified or not in the formula is has level cluster effect or
not (optional), note that this parameter is useful for the function plot
weights Vector of 'prior weights' to be used in the fitting process, by default the weights
are equal to one.
affUser Initial affectation of the individuals in a [data.frame] format, if missing the individ-
uals are randomly assigned to the clusters so it is optional.
timeParametric By default [TRUE] thus parametric on the time. If [FALSE] then only one co-
variate is allowed in the formula and the algorithm used is the k-means.
separateSampling By default [TRUE] it means that the proportions of the clusters are supposed
equal in the classification step, the log-likelihood maximised at each step of the
glmClust

algorithm is \[ \sum_{k=1}^{K} \sum_{y_i \in P_k} \log(f(y_i, \theta_k)), \]
otherwise the proportions of clusters are taken into account and the log-likelihood is \[ \text{latex}. \]

max_itr The maximum number of iterations fixed at 100.
verbose Print the output in the console.

Details

'glmClust' implements an ECM (esperance classification maximisation) type algorithm which assigns the trajectories to the cluster maximising the likelihood. The procedure is repeated until no change in the partitions or no sufficient increase in the likelihood is possible.

'glmClust' also deals with multiple covariates with different level effects, different in each cluster and/or identical for all of them.

The introduction of covariates is possible thanks to 'glm' which fits a generalised linear model and take into account the type of the response (normal, binomial, Poisson ...etc) and the link function.

Several parameters of 'glmClust' are in common with 'glm', like the formula which requires a particular attention by specifying the covariates with a cluster effect, for e.g. clust(T1+T2+...+Tn), the covariates with an identical effect in each cluster are specified with the keyword pop, for e.g. pop(X1+X2+...+Xn), note that these last covariates are optional.

The data are in the long format and no missing values are allowed.

In the parametric case (timeParametric = TRUE) multiples covariates are allowed, in the non-parametric case only one covariate is allowed.

The algorithm depends greatly on the starting condition, which is obtained by randomly affecting the trajectories to the clusters unless the user introduce his own partition. To obtain better results it is desirable to run the algorithm several times from different starting points, therefore it is preferable to use kmlCov which runs the algorithm several times with different number of clusters.

At the end of the algorithm, an object of class GlmCluster is returned and contains information about the affectation of the trajectories, the proportions, the convergence, ...etc. The main trajectories can be simply visualised by plot(my_GlmCluster_Object).

Value

An object of class GlmCluster.

See Also

kmlCov

Examples

data(artifdata)
res <- glmclust(formula = Y ~ clust(time + time2 + time3) + pop(treatTime),
data = artifdata, ident = 'id', timeVar = 'time', effectVar = 'treatment', nClust = 4)
# the trajectories with indices 0 indicate the ones with a normal treatment, 1 indicate a high dose
# the color indicates the clusters
# the proportions are in the table above the diagram
plot(res)

GlmCluster-class  Class GlmCluster

Description

GlmCluster contains all relevant information about the trajectories obtained and the affectation to the clusters.

Objects from the Class

GlmCluster is used inside glmClust and contain all the information to plot and print the main trajectories.

Slots

formula: Object of class formula.
nClust: The number of clusters.
ident: Name of the 'identity' column in the data.
timeVar: Name of the 'time' column in the data.
time: Numeric Vector of the time.
effectVar: Name of a variable with cluster effect or not.
effect: A variable effect, can be a level cluster effect or not.
model.glm: A glm object.
timeParametric: Object of class logical.
particle: Vector of integer containing the affectation of the individuals to the clusters.
particle.long: Same as particle but with repeated measures corresponding to the number of observations for each individual
proportions: Proportions of individuals (trajectories) affected in each cluster
criteria: A matrix which contains the values of the 'log-likelihood', the 'AIC' (Akaike Information Criterion) and 'BIC' (bayesian information criterion).
converge: An object of class Converge.
nIter: Number of iterations of the algorithm.
for_ggplot: A data.frame containing the time and the typical trajectories.

Methods

plot.GlmCluster-method plot Display the main trajectories.
**Note**

Meant to be used internally.

**See Also**

Classes: [Converge](#).  
Plot: plot(GlmCluster).

---

**GlmCluster-methods**  
*Plot the main trajectories*

---

**Description**

Plot the main trajectories of each cluster and print the proportions of each one of it.

**Methods**

plot(GlmCluster): Plot the main trajectories.

**See Also**

kmlCov glmClust

---

**kmlCov**  
*Clustering longitudinal data from different starting conditions*

---

**Description**

'kmlCov' re-launch the algorithm implemented in [glmClust](#), for clustering longitudinal data (trajectories), several times with different starting conditions and various number of clusters.

**Usage**

```r
kmlCov(formula, data, ident, timeVar, nClust = 2:6,  
nRedraw = 20, family = 'gaussian', effectVar = '',  
weights = rep(1,nrow(data)), timeParametric = TRUE,  
separateSampling = TRUE, max_itr = 100, verbose = TRUE)
```
Arguments

**formula**  A symbolic description of the model. In the parametric case we write for example ‘y ~ clust(time+time2) + pop(sex)’, here ‘time’ and ‘time2’ will have a different effect according to the cluster, the ‘sex’ effect is the same for all the clusters. In the non-parametric case only one covariate is allowed.

**data**  A [data.frame] in long format (no missing values) which means that each line corresponds to one measure of the observed phenomenon, and one individual may have multiple measures (lines) identified by an identity column. In the non-parametric case the totality of patients must have all the measurements at all fixed times.

**nClust**  The number of clusters, at least 2 and at most 26.

**nRedraw**  The number of time the algorithm is re-run with different starting conditions.

**ident**  The name of the column identity.

**timeVar**  Specify the column name of the time variable.

**family**  A description of the error distribution and link function to be used in the model, by default ‘gaussian’. This can be a character string naming a family function, a family function or the result of a call to a family function. (See ‘family’ for details of family functions).

**effectVar**  An effect, can be a level cluster effect or not.

**weights**  Vector of ‘prior weights’ to be used in the fitting process, by default the weights are equal to one.

**timeParametric**  By default [TRUE] thus parametric on the time. If [FALSE] then only one covariate is allowed in the formula and the algorithm used is the k-means.

**separateSampling**  By default [TRUE] it means that the proportions of the clusters are supposed equal in the classification step, the log-likelihood maximised at each step of the algorithm is \( \sum_{k=1}^{K} \sum_{y_i \in P_k} \log(f(y_i, \theta_k)) \), otherwise the proportions of clusters are taken into account and the log-likelihood is \( \sum_{k=1}^{K} \sum_{y_i \in P_k} \log(\lambda_k f(y_i, \theta_k)) \).

**max_itr**  The maximum number of iterations fixed at 100.

**verbose**  Print the output in the console.

Details

The purpose of kmlCov is clustering longitudinal data, as well as glmClust, and automate the procedure of re-launching the algorithm from different starting conditions by specifying nRedraw.

The algorithm depends greatly of the starting conditions (initial affection on the trajectories/individuals), so it is recommended to run the algorithm multiple times in order to explore the space of the solutions.

'kmlCov' return a list of list of GlmClusater, the partitions are compared using as criterion the classification log-likelihood, the higher are the best partitions.
Value

A an object of class KmlCovList.

See Also

glmClust
which_best

Examples

data(artifdata)
res <- kmlCov(formula = Y ~ clust(time + time2), data = artifdata, ident = 'id',
timeVar = 'time', effectVar = 'treatment', nClust = 2:3, nRedraw = 2) #run 2 times for each cluster

KmlCovList-class

Class KmlCovList

Description

KmlCovList is an S4 class which contain a list of GlmCluster objects or a list of list of them.

Objects from the Class

This class is used internally by kmlCov

Slots

list_part: Contain a list or a list of "list" of GlmCluster objects.

Methods

plot Display the main trajectories one by one.

Note

Meant to be used internally.

See Also

Classes: GlmCluster.
Plot: plot(KmlCovList).
log_lik

Calculate the log-likelihood

Description

The log-likelihood is calculated with taking into account the type of data ('gaussian', 'binomial', ... etc) and the link function.

Usage

log_lik(y, n, mu, wt, family, nparam, disp_mod)

Arguments

y Observed values.
n Vector of '1's and same length as y.
mu Predicted values.
wt Weights.
family An object of class family.
nparam Number of parameters of the model.
disp_mod Dispersion of the 'glm' model.

Details

This function calculates the log-likelihood for the exponential family, it uses the 'AIC' function to realise this operation.

Value

The log-likelihood of an individual (trajectory).

Note

Meant to be used internally.
majIndica  
Calculate an indicator vector

Description

Calculate and return an indicator vector.

Usage

majIndica(aff_obs, itrClust)

Arguments

aff_obs  Vector of [integer].
itrClust  Number of current cluster.

Value

An indicator vector of the belonging to a cluster.

Note

Meant to be used internally.

plot-methods  
Plot the main trajectories

Description

Plot the main trajectories of an object of class KmlCovList one by one and ask the user to plot the next diagram.

Methods

plot(KmlCovList): Plot the main trajectories.
predict_clust

Creates a character string expression to calculate the predicted values

Description

Given the covariates and the name of the coefficients corresponding to a given cluster, the function construct a character string, which will be used to calculate the predicted values.

Usage

predict_clust(cov, nomCoef, nom_model)

Arguments

cov       Name of the covariates.
nomCoef   Name of the coefficients.
nom_model Name of the glm model.

Details

To calculate the predicted values in each cluster, we need the values of the covariates in the data and the right coefficients in the ‘glm’ object. To do this we construct an expression which will be evaluated inside glmClust.

Value

A character string of the expression of the predicted values of a given cluster.

Note

Meant to be used internally.

rwFormula

Rewrite the formula with all the covariates

Description

Rewrite a given formula with all the covariates, so do note have to write them all.

Usage

rwFormula(formula, col.names, ident)
Arguments

- `formula`: An object of type 'formula' of the form `y ~ .`
- `col.names`: Name of the columns in the data.
- `ident`: Name of the identity column in the data.

Value

- A 'formula' with all covariates.

Note

Meant to be used internally.

---

seperateFormula | Separate the covariates in a formula

Description

Separate the covariates from a 'formula' with a cluster effect from the ones with an identical effect in each cluster if provided.

Usage

`seperateFormula(formula)`

Arguments

- `formula`: A symbolic description of the model. In the parametric case we write for example `y ~ clust(time*time2) + pop(sex)`, here 'time' and 'time2' will have a different effect according to the cluster, the 'sex' effect is the same for all the clusters. In the non-parametric case only one covariate is allowed.

Details

Given a 'formula' of the form `Y ~ clust(T1 + T2 + ... + pop(X1 + X2 + ...)` or just `Y ~ clust(T1 + ... + T2 + ...)` it returns a list of two or one 'formula' of the form `Y ~ T1 + T2 + ... and ~ X1 + X2 + ...` if provided.

The first element of the list correspond to the covariates with a different effects corresponding to the cluster, the 2nd correspond to covariates having an identical effect in each cluster.

In the non-parametric case only **one** covariate is allowed.

Value

- A list of 1 or 2 [formula].

Note

Meant to be used internally.
which_best

Seek the best partitions

Description

Seek the best partitions in an object of class KmlCovList and return the best one of each fixed number of cluster.

Usage

which_best(kmlcovar, crit = "log-class-likelihood")

Arguments

kmlcovar An object of class KmlCovList.
crit Name of the criterion which have to be optimised, CLL for classification log-likelihood AIC for Akaike information criterion and BIC for Bayesian information criterion.

Value

An object of class GlmCluster or KmlCovList.

See Also

kmlCov

Examples

data(artifdata)
res <- kmlCov(formula = Y ~ clust(time + time2), data = artifdata, ident = 'id', timeVar = 'time', effectVar = 'treatment', nClust = 2:3, nRedraw = 2) # run 2 times the algorithm
best <- which_best(res) # return the best partition of each cluster
plot(best)
Index

+ Topic classes
  GlmCluster-class, 9
+ Topic datasets
  artifdata, 5

addIndic, 3
affect_rand, 4
alias (Converge-class), 5
artifdata, 5

Converge, 9, 10
Converge-class, 5
Converge-methods, 5

family, 7, 13

getNomCoef, 6
glmClust, 3, 4, 7, 9–12, 15
GlmCluster, 8, 9, 12
GlmCluster (GlmCluster-class), 9
GlmCluster-class, 9
GlmCluster-methods, 10

kmlCov, 3, 8, 10, 10, 12, 17
kmlcov (kmlcov-package), 2
kmlcov-package, 2
KmlCovList, 14
KmlCovList (KmlCovList-class), 12
KmlCovList-class, 12

log_lik, 13

majIndica, 14

plot, 7, 12
plot, GlmCluster-method, 9
plot, GlmCluster-method
  (GlmCluster-methods), 10
plot, KmlCovList-method (plot-methods), 14
plot-methods, 14

predict_clust, 15
rwFormula, 15
separateFormula, 16
show, Converge-method
  (Converge-methods), 5
which_best, 3, 12, 17